UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/674,606	09/30/2003	Frank Eliot Levine	AUS920030485US1	4480

35525 7590 01/11/2007

IBM CORP (YA)
C/O YEE & ASSOCIATES PC
P.O. BOX 802333
DALLAS, TX 75380

EXAMINER	
FIEGLE, RYAN PAUL	
ART UNIT	PAPER NUMBER
2183	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
2 MONTHS	01/11/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/674,606
Filing Date: 9/30/03
Appellant(s): LEVINE ET AL.

Theodore D. Fay III
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 11/20/06 appealing from the Office action mailed 6/23/06.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The following are the related appeals, interferences, and judicial proceedings known to the examiner which may be related to, directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal:

Application Serial Number 10/675,831, filed September 20, 2003, Attorney docket Number AUS920030479US1; and

Application Serial Number 10/675,778, filed September 20, 2003, Attorney docket Number AUS920030480US1.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

6,253,338	Smolders	6-2001
2004/0030870	Buser	2-2004
5,987,250	Subrahmanyam	11-1999

FOLDOC: The Free Online Dictionary of Computing. Entry for "jump". 11/14/1998.<http://foldoc.org/foldoc.cgi?jump> Viewed: 7/19/06.

computeruser.com. High-Tech Dictionary. Entry for "Unconditional branch." <http://www.computeruser.com/resources/dictionary/definition.html?lookup=5638> Viewed 7/19/06.

Pfaffenberger, Bryan. Webster's New World Computer Dictionary. 9th Ed. Hungry Minds: 2001. Pg. 52.

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

1. Claims 1, 8, 15 and 22-24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Smolders (US Patent 6,253,338) in view of Buser (USPGPub 2004/0030870).

2. As per claim 1:

Smolders teaches a method in a data processing system for monitoring the execution of a program, the method comprising:

associating instructions for calls and returns in the program with a set of indicators (Smolders: column 4, lines 15-18; column 3, lines 29-33; column 5, lines 20-24; column 5, lines 39-43) (Smolders associates all branches with indicators, this includes calls and returns. The indicators are T, the process information and counters.); and

executing the program using a processor, wherein the set of indicators associated with the instructions causes the processor executing the instructions to generate data on calls and returns in the program (Smolders: column 4, lines 18-21; column 4, lines 60-65).

Smolders does not teach his indicators being in a field associated with each instruction for holding a potential indicator, while Buser does (Buser: Abstract; Figure 1).

Buser states that debugging problems arise in shared memory systems because not all processors are able to honor a breakpoint because they are not aware of it. Buser's method solves these deficiencies (Buser 0002, 0003).

Therefore, it would have been obvious to one of the ordinary skill in the pertinent art that Buser's indicator field would be advantageous when using Smolder's debugging method in a shared memory system.

3. As per claim 8:

Claim 8 recites the system for performing the method of claim 1. Smolders teaches a system for performing his method (Smolders: Figure 2). Therefore, claim 8 is rejected for the same reasons as claim 1.

4. As per claim 15:

Claim 15 recites a computer program product in a computer readable medium for performing the method of claim 1. Smolders teaches a computer program product in a computer readable medium for performing his method (Smolders: column 6, lines 15-25). Therefore, claim 15 is rejected for the same reasons as claim 1.

5. As per claims 22-24:

Claims 1, 8 and 15 further comprising the respective indicators each include an element chosen from the group consisting of a flag, a tag field, a threshold, and a count field (Smolders: column 4, lines 15-18; column 3, lines 29-33; column 5, lines 20-24; column 5, lines 39-43) (T is a flag; the counters are count fields).

6. Claims 2-7, 9-14 and 16-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Smolders (US Patent 6,253,338) in combination with Buser (USPGPub 2004/0030870) as applied to claims 1, 8 and 15 above, and in view of Subrahmanyam (US Patent 5,987,250).

7. Smolders teaches claims 1, 8 and 15 for the reasons listed above.

8. As per claim 2:

Smolders does not teach the following limitations which Subrahmanyam does:

responsive to identifying an instruction in an instruction cache for execution during execution of the program, determining whether an indicator from the set of indicators is associated with the instruction (Subrahmanyam: column 4, lines 9-17; column 4, lines 35-38) (A probe location is marked by a flag indicator which is associated with a location in the instruction code, the location being an instruction, which will inherently come from an instruction cache since Subrahmanyam contains an instruction cache (Subrahmanyam: column 3, lines 60-64).); and

generating an interrupt if the indicator is associated with the instruction, wherein the interrupt causes execution of a program to generate data on the calls and returns in the program (Subrahmanyam: column 4, lines 38-43; column 1, lines 12-17; column 4, lines 2-5) (Subrahmanyam states that various analysis tools are well known in the art which his method can be used for. One of these tools is analyzing calls and returns.).

Smolders explicitly inserts monitor code into the instruction stream (Smolders: column 4, lines 21-26). Subrahmanyam states that this is undesirable for several reasons (Subrahmanyam: column 1, lines 50-65). Subrahmanyam's system collects data throughout regular execution, accumulating the data. Then when a probe location is reached, analysis is done and the data is dumped into an external file. When applying Smolders is applied to Subrahmanyam, data would be collected on calls and returns as normal, but it would accumulated until a probe location is reached, rather than interrupting to a analysis tool within the same process after each branch. Doing

this would allow Smolders to observe call and return behavior without, "affecting program behavior," (Subrahmanyam: column 2, lines 1-3) and would avoid the pitfalls observed by Subrahmanyam (Subrahmanyam: column 1, lines 50-65).

Therefore, it would have been obvious to one of ordinary skill in the pertinent art at the time of the applicant's invention that applying Subrahmanyam to Smolders would allow for studying program behavior without affecting the program behavior.

9. As per claim 3:

The method of claim 1, wherein execution of an instruction associated with an indicator in the set of indicators causes passing of control to a process that records calls and returns (Subrahmanyam: column 4, lines 38-43; column 1, lines 12-17; column 4, lines 2-5).

10. As per claim 4:

The method of claim 1, wherein execution of an instruction associated with an indicator in the set of indicators causes passing of control to a process that identifies a calling routine (Subrahmanyam: column 4, lines 38-49).

11. As per claim 5:

The method of claim 4 further comprising:

associating instructions in the calling routing with the set of indicators (Smolders: column 4, lines 15-18; column 3, lines 29-33; column 5, lines 20-24; column 5, lines 39-43); and

executing the program using a processor, wherein the set of indicators associated with the instructions causes the processor executing the instructions in the

calling routine to generate data on calls and returns in the calling routine (Subrahmanyam: column 4, lines 38-43) (Smolders: column 5, lines 20-24; column 5, lines 39-43).

12. As per claim 6:

The method of claim 1, wherein the set of indicators are located in a shadow memory (Subrahmanyam: column 4, lines 49-53) (Smolders: column 3, lines 29-33).

13. As per claim 7:

The method of claim 1 further comprising:

identifying a called routine (Subrahmanyam: column 5, lines 38-51; Figures 4A-4C, 5).

14. As per claim 9-14:

Claims 9-14 recite the system for performing the method of claims 2-8. Smolders teaches a system for performing his method (Smolders: Figure 2). The same would apply when Subrahmanyam is applied to Smolders. Therefore, claims 9-14 are rejected for the same reasons as claims 2-8.

15. As per claims 16-21:

Claims 16-21 recite a computer program product in a computer readable medium for performing the method of claims 2-8. Smolders teaches a computer program product in a computer readable medium for performing his method (Smolders: column 6, lines 15-25).). The same would apply when Subrahmanyam is applied to Smolders. Therefore, claims 16-21 are rejected for the same reasons as claims 2-8.

(10) Response to Argument

The appellant has made the following argument on page 10 of the appeal brief:

> "Smolders teaches generating an interrupt after each branch instruction. The Office Action asserts that this method would include calls and returns. Based on the assertion, the Office Action concludes that Smolders discloses the claimed feature of, 'storing a respective indicator... associated with each call and return in the program'. However, this interpretation ignores the differences between a branch and a call or return."

A call is just a specific word that refers to a branch that branches into a subroutine. Likewise, a return is a branch that branches out of a subroutine.

High-Tech Dictionary from computeruser.com defines unconditional **branch** as, "a computer program instruction that transfers control of a different part of the program without requiring a decision."

FOLDOC, the free online dictionary of computing defines jump as "(Or 'branch') The term for a goto instruction, usually in a context of machine languages. 'Branch' may be synonymous with 'jump', or may refer to jumps that depend on a condition."

Webster's New World Computer Dictionary, 9th Ed., defines branch as "3. In programming, a type of control structure that **routes program execution to a subroutine**."

Therefore, calls and returns, both, **are** in fact branches, though special kinds of branches. However, some, such as Webster's New World Computer Dictionary, explicitly consider a branch to be a call.

The appellant argues on pages 11 and 12 of the appeal brief that calls and returns differ from branch instructions because calls and returns infer state management and address saving. However, it should be noted that there are some instruction set architectures that do not have dedicated call and return instructions that

do this; it is required to do it manually. In addition, there are no set rules for what constitutes a call or return. Many architectures have differing calling conventions, meaning that sometimes it is up to the caller to save the state while sometimes it is the callee's responsibility. Further, the appellant did not disclose any explicit definitions for either call or return within the specification.

Smolders does not explicitly state that his method is used for calls and returns, but merely branches (column 4, lines 15-18). However, since it has been shown that calls and returns are unquestionably types of branches, it would have been obvious to one of ordinary skill in the pertinent art that Smolder's method would be applicable to calls and returns. Little, if any, modifications would need to be made to the present implementation.

It should also be noted that each branch in Smolders causes an interrupt to collect the data. By definition, an interrupt will make a call to a subroutine, in this case to collect the data, and then return to where the program was interrupted. The claim limitation in question states, "storing an indicator in the indicator location associated with each call and return in the program." A broad and reasonable interpretation of the claim would lead one of ordinary skill in the art to come to the conclusion that since each branch in Smolders will cause a call and return associated with it, and the data generation within the subroutine will be associated with these calls and returns, Smolders interrupt caused by encountering the branch will fulfill the limitation of "storing an indicator in the indicator location associated with each call and return in the program" in addition to the reasons stated above.

The appellant has made the following argument on page 12 of the appeal brief:

> "The information generated by the method recited in Claim 1 is specifically 'data on calls and returns;' the information is not data generation that is triggered by a call or return."

The appellant is using the following interpretation: "...executing the instructions to generate data **about** calls and returns in the program." On the other hand, the examiner has taken the interpretation, "...executing the instructions to generate data **upon encountering** calls and returns in the program." Both are acceptable interpretations of the use of "on" within the claim. Neither interpretation is precluded by the remainder of the claims or explicit excerpts from the specification.

It should further be noted that the secondary reference of Subrahmanyam, used in the rejection of other claims, explicitly teaches generating data ***about*** calls and returns.

The appellant has made the following argument on page 13 of the appeal brief:

> "However, the proposed motivation does not actually exist because Smolders is not concerned with honoring a common breakpoint among multiple processors."

The appellant is arguing that simply because Smolders is not explicitly set in a shared memory system, there would be no motivation to combine with Buser.

Smolders states that his invention, "relates in general to a method and system for data processing and in particular to an improved method and system for counter level tracing." Figure 1 of the invention states that Smolders **comprises**, not consists of, a system processor unit 12, a display 18, a keyboard 20, and a mouse 22.

Shared memory systems are very common these days. In fact, any home computer that is bought from any store today (which contains technology that was patented several years ago, before the time of invention of the appellant) will contain a shared memory system. This is because of the development of dual core chips such as the Pentium D, Core and X2. All of these chips basically have multiple processors contained on a single chip, and therefore must share memory.

If Smolders were implemented at the time of the appellant's invention, even in the absence of Buser, it is likely that it would ***have*** to be a shared memory system, and therefore the motivation to combine with Buser is imminently there.

The appellant has made the following argument on page 16 of the appeal brief:

> "Because the references address completely distinct problems, one of ordinary skill would have no reason to combine or otherwise modify the references to achieve the invention of Claim 1."

Both Smolders and Buser have classifications to 712/227, specialized instruction processing in support of testing, debugging and emulating. Further, proper motivation has been shown within the rejection that it would be beneficial to apply Buser to Smolders since there is no other way for multiple processors to see the indicators of Smolders in a shared memory system. As stated above, the implementation and benefits of a shared memory system are well known and therefore one of ordinary skill would readily be motivated to combine Buser to Smolders.

The appellant has made the following argument on page 18 of the appeal brief:

> "Under the standards of Graham v. John Deere, the examiner is required to provide a proper teaching, suggestion, or motivation to combine all of the reference together as a whole. By ignoring Buser the Office Action has failed to comply with this requirement."

The application of Subrahmanyam to Smolders and Buser is inconsequential to how the combination of Smolders and Buser would perform. The addition of Buser to Smolders simply moves the indicators from hardware registers and the like to a field associated with each of the instructions. Subrahmanyam was brought in to avoid the adverse affects of inserting monitor code directly into the instruction stream (Subrahmanyam: column 1, lines 50-65). Therefore, when Subrahmanyam is applied to Smolders and Buser as opposed to only Smolders, the only thing that will change is where the data is obtained during Subrahmanyam's data collection. In Smolders and Buser, this will come from the indicator instruction field, while in Smolders it would come from the hardware registers. Therefore, the motivation to apply Subrahmanyam to Smolders and Buser would be the same as the motivation to apply Subrahmanyam to Smolders.

The appellant has made the following argument on page 20 of the appeal brief:

> "Because the references address completely distinct problems, one of ordinary skill would have no reason to combine or otherwise modify the references to achieve the invention of Claim 2."

Smolders, Buser and Subrahmanyam all deal with specialized instruction processing in support of testing, debugging and emulating. Further, proper motivation has been shown within the rejection that it would be beneficial to apply Subrahmanyam to Buser and Smolders since the combination of Smolders and Buser would insert

monitor code directly into the instruction stream which has been found to be disadvantageous (Subrahmanyam: column 1, lines 50-65).

Section B.2.ii.a in the appeal brief is incorrectly labeled as pertaining to claim 3. The section actually refers to limitations of claim 5.

The appellant has made the following argument on page 23 of the appeal brief:

> "The fact that a call graph based profiling tool exists provides no indication that indicator locations be associated with instructions in the calling routine."

If the indicators were not associated with instructions in the calling routine, there would be no way for the calling graph to see nested and recursive calls, which is the entire point of the graph.

(11) Related Proceeding(s) Appendix

The following related applications are currently under appeal. No decisions have been made in the related applications.

Application Serial Number 10/675,831, filed September 20, 2003, Attorney docket Number AUS920030479US1; and

Application Serial Number 10/675,778, filed September 20, 2003, Attorney docket Number AUS920030480US1.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Ryan Fiegle

Conferees:

Lynne Browne

LYNNE H BROWNE
APPEAL PRACTICE SPECIALIST
TECHNOLOGY CENTER 2100

Eddie Chan

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100